

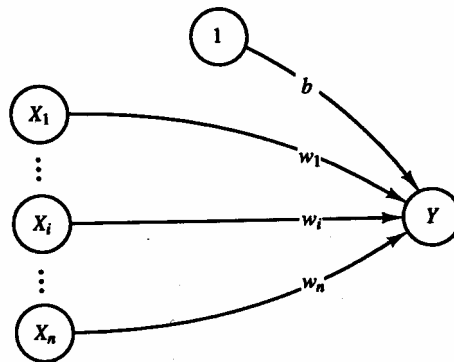
BAB VB

PERSEPTRON & CONTOH

Model JST perseptron ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969). Model ini merupakan model yang memiliki aplikasi dan pelatihan yang lebih baik pada era tersebut.

5B.1 Arsitektur Jaringan

Arsitektur jaringan perseptron mirip dengan arsitektur jaringan Hebbian



Jaringan terdiri dari beberapa unit masukan dan sebuah bias, dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasinya bukan merupakan fungsi biner atau bipolar, tetapi memiliki kemungkinan nilai -1, 0 dan 1:

Untuk suatu harga threshold θ yang ditentukan :

$$f(net) = \begin{cases} 1 & \text{jika } net > \theta \\ 0 & \text{jika } -\theta \leq net \leq \theta \\ -1 & \text{jika } net < -\theta \end{cases}$$

Secara geometris, fungsi aktivasi membentuk 2 garis sekaligus (memisahkan daerah positif dan daerah negatif).

Garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan :

$$x_1w_1 + \dots + x_nw_n + b > \theta$$

Garis pemisah antara daerah negatif dan daerah nol memiliki pertidaksamaan :

$$x_1w_1 + \dots + x_nw_n + b < -\theta$$

5B.2 Algoritma Pelatihan Perseptron

Misalkan :

s vektor masukan dan t adalah target keluaran

α adalah laju pembelajaran/pemahaman (learning rate) yang ditentukan

θ adalah threshold yang ditentukan

Algoritma sbb:

0. Inisialisasi semua bobot dan bias (umumnya $w_i = b = 0$)

Set laju pembelajaran α ($0 < \alpha \leq 1$) (untuk penyederhanaan set $\alpha = 1$)

1. Selama kondisi berhenti bernilai FALSE atau selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target ($y \neq t$), lakukan langkah-langkah 2 – 6

2. Untuk setiap pasangan (s,t), kerjakan langkah 3 – 5:

3. Set aktivasi unit masukan $x_i = s_i$ ($i = 1, \dots, n$)

4. Hitung respon untuk unit output:

$$\text{net} = \sum_i x_i w_i + b$$

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases}$$

5. Perbaiki bobot dan bias pola jika terjadi kesalahan

Jika $y \neq t$, maka:

$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w_i \quad (i = 1 \dots n); \quad \text{dengan } \Delta w_i = \alpha t x_i$$

$$b (\text{baru}) = b (\text{lama}) + \Delta b \quad \text{dengan } \Delta b = \alpha t$$

Jika $y = t$ maka:

$$w_i (\text{baru}) = w_i (\text{lama})$$

$$b (\text{baru}) = b (\text{lama})$$

6. Test kondisi berhenti: jika tidak terjadi perubahan bobot pada (i) maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE

Ada beberapa hal yang perlu diperhatikan dalam algoritma tersebut:

- a. Iterasi terus dilakukan hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola). Iterasi TIDAK berhenti setelah semua pola dimasukan seperti yang terjadi pada model Hebbian
- b. Pada langkah 5, perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan (keluaran jaringan \neq target). Perubahan tersebut merupakan hasil kali unit masukan dengan target dan laju pembelajaran. Perubahan bobot hanya akan terjadi jika unit masukan $\neq 0$
- c. Kecepatan iterasi ditentukan oleh laju pembelajaran α (dengan $0 < \alpha \leq 1$) yang dipakai. Semakin besar α , semakin sedikit iterasi yang diperlukan. Akan tetapi jika α terlalu besar, maka akan merusak pola yang sudah benar sehingga pembelajaran menjadi lambat

Algoritma pelatihan perseptron lebih baik dibandingkan model Hebbian karena:

- a. Setiap kali pola dimasukkan, hasil keluaran jaringan dibandingkan dengan target yang sesungguhnya. Jika terdapat perbedaan, maka bobot akan dimodifikasi. Jadi tidak semua bobot selalu dimodifikasi dalam setiap iterasinya
- b. Modifikasi bobot tidak hanya ditentukan oleh perkalian antara target dan masukan, tetapi juga melibatkan suatu laju pembelajaran yang besar dapat diatur.
- c. Pelatihan dilakukan berulang-ulang untuk semua kemungkinan pola yang ada hingga jaringan dapat mengerti polanya (ditandai dengan samanya semua keluaran jaringan dengan target keluaran yang diinginkan). Satu siklus pelatihan yang melibatkan semua pola disebut ***Epoch***. Dalam jaringan Hebbian, pelatihan hanya dilakukan dalam satu Epoch saja.
- d. Teorema konvergensi perseptron menyatakan bahwa apabila ada bobot yang tepat, maka proses pelatihan akan konvergen ke bobot yang tepat tersebut

Contoh 1:

Buatlah perseptron untuk mengenali fungsi logika AND dengan masukan dan keluaran bipolar. Untuk inialisasi, gunakan bobot dan bias awal = 0, $\alpha = 1$, dan threshold $\theta = 0$

Penyelesaian

Tabel masukan dan target fungsi logika AND dengan masukan dan keluaran bipolar

Masukan			Target
x_1	x_2	1	t
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Untuk threshold $\theta = 0$, maka fungsi aktivasinya:

$$f(net) = \begin{cases} 1 & \text{jika } net > 0 \\ 0 & \text{jika } net = 0 \\ -1 & \text{jika } net < 0 \end{cases}$$

Iterasi untuk seluruh pola yang ada disebut EPOCH. Tabel berikut menunjukkan hasil pada **epoch pertama**

Masukan	target			Perubahan bobot	Bobot Baru
(x_1 x_2 1)	t	net	y = f(net)	(Δw_1 Δw_2 Δb)	(w_1 w_2 b)
Inialisasi					(0 0 0)
(1 1 1)	1	0	0	(1 1 1)	(1 1 1)
(1 -1 1)	-1	1	1	(-1 1 -1)	(0 2 0)
(-1 1 1)	-1	2	1	(1 -1 -1)	(1 1 -1)
(-1 -1 1)	-1	-3	-1	(0 0 0)	(1 1 -1)

Pada input pertama $(x_1 \ x_2 \ 1) = (1 \ 1 \ 1)$. Nilai net dihitung berdasarkan bobot yang sudah ada sebelumnya yaitu $(w_1 \ w_2 \ b) = (0 \ 0 \ 0)$. Maka net =

$$\sum_i x_i w_i + b = 1(0) + 1(0) + 0 = 0, \text{ sehingga } f(\text{net}) = f(0) = 0.$$

Keluaran jaringan ($f(\text{net})=0$) TIDAK SAMA dengan target yang diinginkan (dalam iterasi ini target = 1), maka bobot harus diubah menggunakan rumus:

$$\Delta w_i = \alpha t x_i = t x_i \text{ (karena } \alpha = 1\text{)}. \text{ Bobot baru} = \text{bobot (lama)} + \Delta w_i$$

Input pola kedua dan seterusnya dihitung dengan cara yang sama.

Pada pola terakhir $(x_1 \ x_2 \ 1) = (-1 \ -1 \ 1)$, nilai $f(\text{net}) = -1$ yang sama dengan targetnya. Maka bobot tidak diubah. Hal ini dinyatakan dengan kondisi $\Delta w_i = 0$

Garis pemisah pola terbentuk dari persamaan:

$$x_1 w_1 + x_2 w_2 + b = 0 \text{ (karena } \theta = 0\text{)}, \text{ jadi terbentuk sebuah garis saja}$$

Persamaan garis pemisah untuk setiap pola hasil iterasi:

Masukan	Bobot baru	Persamaan garis
$(x_1 \ x_2 \ 1)$	$(w_1 \ w_2 \ b)$	
$(1 \ 1 \ 1)$	$(1 \ 1 \ 1)$	$x_1 + x_2 = -1$
$(1 \ -1 \ 1)$	$(0 \ 2 \ 0)$	$x_2 = 0$
$(-1 \ 1 \ 1)$	$(1 \ 1 \ -1)$	$x_1 + x_2 = 1$
$(-1 \ -1 \ 1)$	$(1 \ 1 \ -1)$	$x_1 + x_2 = 1$

Mengingat tidak semua $f(\text{net})$ pada tabel di atas sama dengan t , maka iterasi dilanjutkan pada **epoch kedua**. Semua pola kembali dimasukkan ke jaringan dengan menggunakan bobot terakhir yang diperoleh sebagai bobot awalnya.

Diperoleh hasil iterasi pada tabel di bawah berikut:

Masukan	target			Perubahan bobot	Bobot Baru
$(x_1 \ x_2 \ 1)$	t	net	$y = f(\text{net})$	$(\Delta w_1 \ \Delta w_2 \ \Delta b)$	$(w_1 \ w_2 \ b)$
Bobot yang diperoleh dari epoch pertama					$(1 \ 1 \ -1)$
$(1 \ 1 \ 1)$	1	1	1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(1 \ -1 \ 1)$	-1	-1	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(-1 \ 1 \ 1)$	-1	-1	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$
$(-1 \ -1 \ 1)$	-1	-3	-1	$(0 \ 0 \ 0)$	$(1 \ 1 \ -1)$

Dalam iterasi di atas, untuk semua pola nilai $f(\text{net}) = t$, sehingga tidak dilakukan perubahan terhadap bobot. Karena $f(\text{net}) = t$ untuk semua pola maka jaringan sudah mengenal semua pola sehingga iterasi dihentikan

Latihan:

1. Ulangi contoh di atas, tapi menggunakan masukan biner dan keluaran bipolar. Gunakan $\alpha = 1$ dan threshold $\theta = 0,2$.
2. Diketahui perseptron dengan 3 masukan x_1, x_2, x_3 , sebuah bias dan sebuah keluaran bipolar. Carilah bobot yang akan mengenali pola sebagai berikut: target keluaran bernilai 1 apabila semua masukan bernilai 1, dan target bernilai -1 apabila tepat salah satu dari masukan bernilai 0 (tidak diketahui bagaimana target apabila ada dua atau lebih masukan yang bernilai 0). Gunakan bobot awal dan bias = 0, $\alpha = 1$ dan threshold $\theta = 0,1$.

5B.3 Pengenalan Pola Karakter

Perseptron dapat pula digunakan untuk mengenali pola karakter. Dengan berbagai pola masukan yang menyerupai huruf-huruf alfabet, perseptron dapat dilatih untuk mengenalinya.

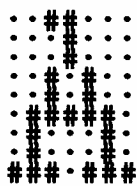
5B.3.1 Pengenalan Sebuah Pola karakter

Algoritma untuk mengenali apakah pola masukan yang diberikan menyerupai sebuah karakter tertentu (misal huruf "A") atau tidak adalah sebagai berikut:

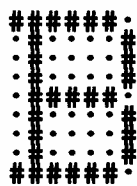
1. Nyatakan tiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut
2. Berikan nilai target = +1 jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target = -1
3. Berikan inisialisasi bobot, bias, laju pembelajaran dan threshold
4. Lakukan proses pelatihan perseptron seperti contoh 1 di atas

Contoh 2:

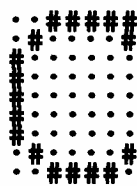
Diketahui 6 buah pola seperti tampak pada gambar berikut:



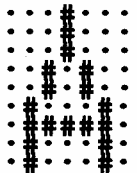
Pola 1



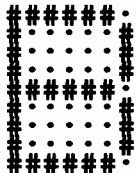
Pola 2



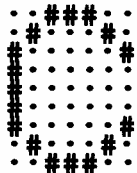
Pola 3



Pola 4



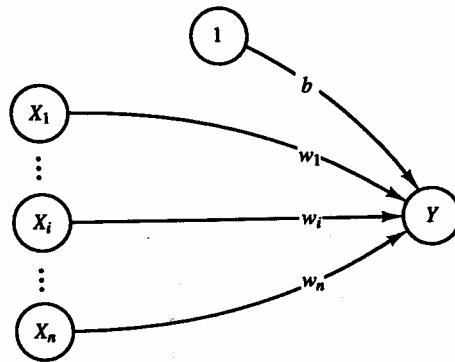
Pola 5



Pola 6

Penyelesaian:

Untuk menentukan vektor masukan, tiap titik dalam pola diambil sebagai komponen vektor. Jadi tiap vektor masukan memiliki $9 \times 7 = 63$ komponen. Titik dalam pola yang bertanda "#" diberikan nilai = 1, dan titik dalam pola bertanda "." diberikan nilai = -1. Pembacaan dimulai dari kiri ke kanan, dimulai dari baris paling atas



Arsitektur jaringan perseptron dengan $n = 63$

Vektor masukan pola 1 adalah

```

-1 -1 1 1 -1 -1 -1
-1 -1 -1 1 -1 -1 -1
-1 -1 1 1 -1 -1 -1
-1 -1 1 -1 1 -1 -1
-1 -1 1 -1 1 -1 -1
-1 1 1 1 1 1 -1
-1 1 -1 -1 -1 1 -1
-1 1 -1 -1 -1 1 -1
1 1 1 -1 1 1 1

```

dst. untuk pola 2 – pola 6

Target bernilai = 1 jika pola masukan menyerupai huruf "A". Jika tidak maka target bernilai = -1. Pola yang menyerupai huruf A adalah pola 1 dan pola 4. Kedua pola memiliki target = 1. Pola yang lain memiliki target = 0

Maka perseptron yang dipakai untuk mengenali huruf "A" (atau bukan "A") memiliki 63 unit masukan, sebuah bias dan sebuah unit keluaran.

Misalkan semua bobot awal dan bias diambil = 0, $\alpha = 1$ dan threshold $\theta = 0,5$. Pelatihan dilakukan dengan cara memasukan 63 unit masukan (atau sebuah pola huruf).

► Hitung net = $\sum_{i=1}^{63} x_i w_i + b$;

► Hitung fungsi aktivasi dengan persamaan:

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > 0,5 \\ 0 & \text{jika } -0,5 \leq \text{net} \leq 0,5 \\ -1 & \text{jika } \text{net} < -0,5 \end{cases}$$

- ▶ Jika $f(\text{net})$ tidak sama dengan target, bobot diubah
- ▶ Proses pelatihan dilakukan terus hingga semua keluaran jaringan sama dengan targetnya

5B.3.2 Pengenalan Beberapa Pola karakter

Pengenalan beberapa pola karakter sekaligus (misal huruf A, bukan A, B, bukan B, dst) dilakukan dengan cara menggabungkan beberapa model perseptron pada bab 5B.3.1. Jadi ada beberapa unit keluaran sekaligus.

Setiap unit masukan dihubungkan dengan setiap unit target. Bobot penghubung dari unit x_i ke y_j adalah w_{ji} (perhatikan bagaimana indeks dibuat). Demikian juga dengan bias dihubungkan dengan semua unit keluaran dengan bobot masing-masing b_1, b_2, \dots, b_m

Algoritma pelatihannya sbb:

1. Nyatakan tiap pola masukan sebagai vektor bipolar yang elemennya adalah tiap titik dalam pola tersebut
2. Berikan nilai target $t_j = 1$ jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target $t_j = -1$ ($j = 1, \dots, m$)
3. Berikan inisialisasi bobot, bias, laju pembelajaran dan threshold
4. Lakukan proses pelatihan perseptron seperti contoh 1 di atas

a. Hitung respon unit keluaran ke- j : $\text{net}_j = \sum_{i=1}^{63} x_i w_{ji} + b_j$

$$y_j = f(\text{net}_j) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases}$$

- b. Perbaiki bobot pola yang mengandung kesalahan ($y_j \neq t_j$) menurut persamaan:
 $w_{ji}(\text{baru}) = w_{ji}(\text{lama}) + \alpha t_j x_i$
 $b_j(\text{baru}) = b_j(\text{lama}) + \alpha t_j$
- c. Lakukan langkah 4a – b terus menerus hingga $y_j = t_j$ ($j = 1, \dots, m$)

Contoh 3:

Diketahui 6 buah pola masukan seperti pada gambar contoh 2
Buatlah model perseptron untuk mengenali pola "A", "B" dan "C".

Penyelesaian:

Mula-mula dibuat 6 buah vektor masukan seperti contoh 2 di atas. Ada 3 buah vektor keluaran yang masing-masing menyatakan bahwa pola menyerupai huruf "A", "B" dan "C". Vektor target tampak pada tabel berikut:

Pola masukan	t_1	t_2	t_3
Pola 1	1	-1	-1
Pola 2	-1	1	-1
Pola 3	-1	-1	1
Pola 4	1	-1	-1
Pola 5	-1	1	-1
Pola 6	-1	-1	1

Selanjutnya iterasi dapat dibuat seperti pada contoh-contoh di atas, dengan beberapa perubahan sbb:

1. Vektor masukan terdiri dari 63 elemen $x_1 \dots x_{63}$ dan sebuah bias
2. Ada 3 target t_1, t_2 dan t_3
3. Ada 3 kolom net yaitu net_1, net_2, net_3 masing-masing merupakan hasil kali bobot dengan vektor masukan
4. Ada 3 buah y yang merupakan fungsi aktivasi ketiga kolom net. $y_1 = f(net_1)$, $y_2 = f(net_2)$, $y_3 = f(net_3)$
5. Ada 3 * 63 buah kolom perubahan bobot yaitu
 $\Delta W_{1,1}, \dots, \Delta W_{1,63}; \Delta W_{2,1}, \dots, \Delta W_{2,63}; \Delta W_{3,1}, \dots, \Delta W_{3,63}$
6. Ada 3*63 bobot
 $W_{1,1}, \dots, W_{1,63}; W_{2,1}, \dots, W_{2,63}; W_{3,1}, \dots, W_{3,63}$